ポリゴン情報の最小トライアングルストリップ化 (21世紀の数理計画 : アルゴリズムとモデリング)

AUTHOR(S):

木幡, 周治; 久野, 誉人; 徳永, 隆治; 長野, 寛

CITATION:

木幡, 周治 ...[et al]. ポリゴン情報の最小トライアングルストリップ化 (21世紀の数理計画 : アルゴリズムとモデリング). 数理解析研究所講究録 2010, 1676: 209-222

ISSUE DATE:

2010-04

URL:

http://hdl.handle.net/2433/141249

数理解析研究所講究録
第1676巻 2010年 209-222

ポリゴン情報の最小トライアングルストリップ化

筑波大学大学院 システム情報工学研究科
木幡周治 (Shuji Kohata)*, 久野誉人 (Takahito Kuno),
徳永隆治 (Ryuji Tokunaga), 長野 寛 (Hiroshi Nagano)
Graduate School of Systems and Information Engineering,
University of Tsukuba

概 要

3次元物体の形状を表すポリゴンモデル情報の最小トライアングルストリップ化について，組合せ最適化の手法を用いて研究した．近年ではポリゴンモデルの利用は多岐に渡り，重要性が高まっている．またより詳細で滑らかな形状を表す需要のためモデル毎のポリゴン数は増加していく傾向にあり，そのためポリゴン数の多いモデルを上手く扱う手法が研究されている．本稿では，トライアングルリストで与えられたポリゴンモデル情報をなるべく小さい縮退三角形を利用したトライアングルストリップに変換する問題を組合せ最適化問題として定義し，それに対する実用的な解法のひな形を提案する．

1 はじめに

ポリゴンとは，3次元グラフィクスの分野で良く使われる概念で，多角形の集合体で立体物を表す表現手法である (図1)．近年では，工業デザインからアミューズメントに至る様々な分野で3次元グラフィクス技術の需要が増加している．中でもポリゴンモデリングの技術は複雑な形状を少ない計算量で表現できるため，中心的な技術となっている．

トライアングルストリップはポリゴンモデルの位相情報（どの頂点を使って三角形が作られるか）を効率良く表現することが可能な手法で，データサイズを小さくでき，パイプライン化に適していることから，モデルのデータの保持や，レンダリングの際に利用される．

トライアングルストリップには利用する目的やハードウェアの仕様等によって細かい様々なバリエーションが存在する [10, 12]．裏表が区別される場合とそうでない場合があり，ストリップの方向等をフラグによって制御する場合もある [1]．また，良いトライアングルストリップという指標にも利用や条件に応じて様々なものがある．同じ形状を表現できるならなるべく小さいデータサイズで表現できるほうがよいし，ハードウェアによってレンダリングされる際には頂点キャッシュを考慮し [7]，ハードウェアの仕様に適しているほうが良い．データ圧縮等の目的で利用する場合には，データ列に規則性があり，予測符号化が効きやすいほうが良い．

それに対してトライアングルストリップ化の手法に関しては，実装的なものを中心に実装・研究されている [3, 7, 9]．また，前の三角形との位置関係をフラグで管理できる一般化トライアング

*kohata@syou.cs.tsukuba.ac.jp

図 1: ポリゴンによる表現の例. Stanford Bunny[11]

ルストリップに関しては，巡回セールスマン問題 [8] に直接変換できるため，良く研究が行われている [2, 6].

本研究ではポリゴンモデルの位相情報のトライアングルストリップ化手法に関して，特にデータサイズのなるべく小さい（一般化でない）トライアングルストリップの生成に着目し，組合せ最適化問題として捉えて解析，実用的なトライアングルストリップ化アルゴリズム作成に役立てたい.

2 ポリゴン情報の表現とトライアングルストリップ

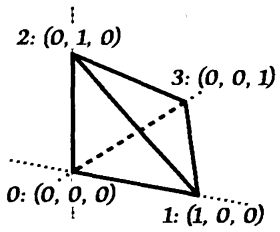
本節では，ポリゴンモデルの情報がどのように表現されるか，また，本稿で扱っていくトライアングルストリップがどのようなものかを解説していく．言葉の定義等は OpenGL [12] や Direct3D [10] で使われているものを使用している.

2.1 ポリゴン情報の表現

ポリゴン情報は，物体の形状を複数の多角形の集合体により表したものである．多角形の種類としては，分かりやすさから三角形や四角形が使われるが，全ての多角形は三角形に分割可能で，3 頂点は同一平面上にあることが保証されるため，理論上や描画システム上では三角形のみの集合とする場合が多い．ここでも，そのような三角形の集合体であるとする.

ポリゴン情報は各頂点の座標値を表す幾何情報と頂点の接続関係を表す位相情報に分けて考え

ることが可能である（図2）．この場合，位相情報はトライアングルリストとして表現される．



v_0	$(0,0,0)$
v_1	$(1,0,0)$
v_2	$(0,1,0)$
v_3	$(0,0,1)$

t_0	0	1	2
t_1	2	1	3
t_2	2	3	0
t_3	0	3	1

図 2: ポリゴン情報の表現例 (四面体)．左が各頂点の座標，右がトライアングルリスト

また，三角形の集合を効率良く表現するために，トライアングルストリップ，トライアングルファンという表現法が存在する（図3）．トライアングルストリップとトライアングルファンは共に頂点列で三角形の集合を表し，トライアングルストリップは頂点列中の連続する 3 頂点で三角形を，トライアングルファンは頂点列の最初の点と他の連続する 2 頂点で三角形を生成する．また，特に有効であるトライアングルストリップに関しては，以降で解説していく．

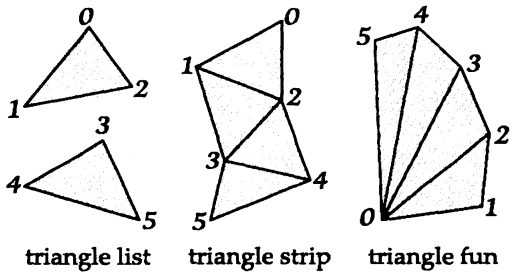
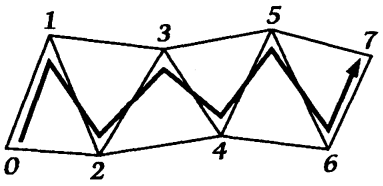


図 3: 応用的な位相情報の表現

2.2 トライアングルストリップ

トライアングルストリップは三角形の集合を効率良く表現することの可能な手法である．頂点の列で与えられ，列中の連続する 3 頂点で三角形を表現する（図4）．

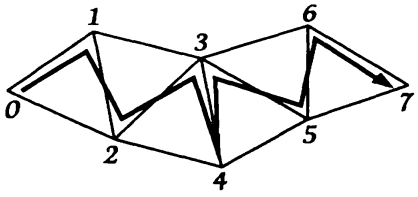
トライアングルストリップはその単純さと次の動作の予測のしやすさからパイプライン化に適しており，専用ハードウェアによるリアルタイムレンダリングやオンライン圧縮・伸張等に有効利用することができる．



triangle strip :
0 1 2 3 4 5 6 7

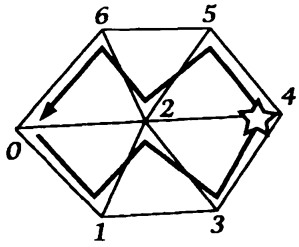
図 4: トライアングルストリップ

トライアングルストリップをそのまま使う場合，帯状の三角形の集合のみしか表現することができないが，表現したい三角形の間に頂点重複を含み面が描画されない縮退三角形を挟むことで，様々な配置の三角形の集合を表現することができる．図5〜図8に縮退三角形を使った三角形群表現の一通りの例を示す．"x 縮退" は縮退三角形が x 個出現していることを示し，図中の星印は同じ頂点が2回連続で現れることを示している．



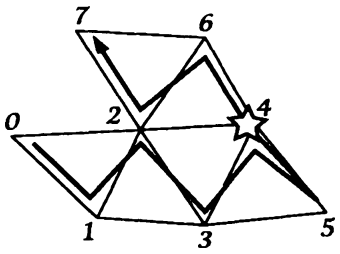
triangle strip :
0 1 2 3 4 3 5 6 7

図 5: トライアングルストリップの縮退：1 縮退の曲がるパターン



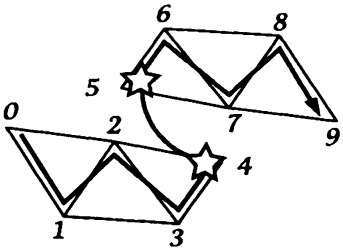
triangle strip :
0 1 2 3 4 4 5 2 6 0

図 6: トライアングルストリップの縮退：2 縮退の折り返すパターン



triangle strip :
0 1 2 3 4 5 4 4 6
2 7

図 7: トライアングルストリップの縮退：3 縮退の鋭く折り返すパターン



triangle strip :
0 1 2 3 4 4 5 5 6 7
8 9

図 8: トライアングルストリップの縮退：4 縮退の跳ぶパターン

3 最小トライアングルストリップ化問題

本節では，トライアングルリストによって与えられたポリゴンモデル情報を，同様の三角形の集合を表現する最小の1本のトライアングルストリップに変換する問題を組合せ最適化問題 [8] として解析していく．

3.1 問題の定義

組合せ最適化問題として定義しやすくするために，単純な問題を取り出し，細かいバリエーションへの対応は別に考えることにする．ここではトライアングルストリップは縮退三角形を利用し裏表の区別の無いものとし，トライアングルストリップの評価はストリップ長のみで行う．

また，トライアングルストリップのストリップ長は，

(表現する三角形の数) + (縮退三角形の個数) + 2

で表現できるため，

最小化 ： トライアングルストリップ中の縮退三角形の個数
制約条件 ： トライアングルストリップが，与えられたトライアングルリストを過不足なく表している．

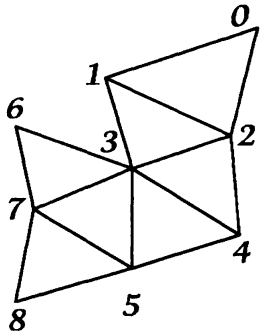
というような組合せ最適化問題として考えることができる．

3.2 より扱いやすい表現

前節の問題定義の制約条件は，トライアングルストリップの長さが可変なために，そのままでは非常に扱いにくい．制約条件を扱いやすくするために，トライアングルストリップ中の，三角形の出現順と，各三角形の頂点出現順について考えていく．

まず，与えるトライアングルリストの三角形数を n として，トライアングルリストが n 行 3 列の頂点番号の行列 T で表現されているとする (図 9)．

制約条件を満たしているようなトライアングルストリップを考えると，全ての三角形が過不足無く出現するので，それを順番に並べることができる．また，各三角形について，その三角形を構成する 3 頂点も出現順に並べることができる．結果として n 行 3 列の行列ができ，それは T の行を並びかえたのち，各行毎に列を並びかえたものである．これを T' とする (図 10)．



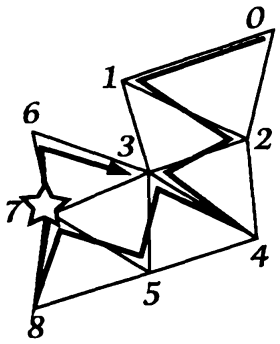
T :

t_0	0	1	2
t_1	1	3	2
t_2	2	3	4
t_3	3	5	4
t_4	3	6	7
t_5	3	7	5
t_6	5	7	8

図 9: トライアングルリスト

ここで逆に，T' を生成するトライアングルストリップのうち最小のものがどんなものかを考える．これは，T' の上下に連続した行の関係，つまり連続して現れる三角形の関係を使って求めていくことが可能である．表 1 に連続する 2 つの三角形の一通りの接続例を示す．表の接続三角形の下線の引かれた部分が等しい時にその縮退三角形数での接続が可能である．

ここまでの議論により，最小のトライアングルストリップにも対応する T' が存在し，その T' から同様のストリップを求めることができることが言え，三角形の出現順と各三角形中の頂点出現順を変数として扱っても，最小トライアングルストリップを求めるのに問題がないことが分かる．それをふまえて，以下にもう一度組合せ最適化問題としてまとめておく．



triangle strip :
0 1 2 3 4 3 5 7 8 7
7 6 3

T' :

t'_0	0	1	2
t'_1	1	2	3
t'_2	2	3	4
t'_3	4	3	5
t'_4	3	5	7
t'_5	5	7	8
t'_6	7	6	3

図 10: トライアングルストリップからの生成される頂点行列

表 1: 連続する 2 つの三角形の接続例

接続三角形		接続部の頂点列	縮退三角形数
0 1 2	1 2 3	0 1 2 3	0 : 直進
0 1 2	2 1 3	0 1 2 1 3	1 : 曲がり
0 1 2	2 3 4	0 1 2 2 3 4	2 : 折り返し
0 1 2	1 3 4	0 1 2 1 1 3 4	3 : 鋭い折り返し
0 1 2	3 2 4	0 1 2 2 3 2 4	3 : 鋭い折り返し
0 1 2	3 4 5	0 1 2 2 3 3 4 5	4 : 跳び

最小化 : $\sum_{k=0}^{n-2} c(\ t'(k,1), t'(k,2), t'(k+1,0), t'(k+1,1)\ )$

制約条件 : T' は T の行を並びかえたのち，各行毎に列を並びかえたものである

但し，

$t'(i,j)$: T' の i 行 j 列（0 から数えて）

$$c(x_1,x_2,y_1,y_2) = \begin{cases} 0 & \text{if } x_1=y_1 \wedge x_2=y_2 \\ 1 & \text{if } x_2=y_1 \wedge x_1=y_2 \\ 2 & \text{if } x_2=y_1 \wedge x_1\neq y_2 \\ 3 & \text{if } x_1=y_1 \wedge x_2\neq y_2 \\ & \quad \vee\, x_1\neq y_1 \wedge x_2=y_2 \\ 4 & \text{otherwise} \end{cases}$$

4 計算複雑性について

本節では第 3 節で定義した問題を厳密に解くことの難し さについて議論する．次節以降で，最適化解が求まるとは限らない手法を使っていく 1 つの理由になっている．

4.1 NP 困難性とその証明手順

クラス NP 困難 [5] とは，計算複雑性の理論において重要な概念で，入力データサイズの多項式で表現される時間で解けることが保証できそうもないとされている問題のクラスである．

ある問題 B の NP 困難性の証明のためには，NP 完全 [5] であると知られている問題 A を用意し，その問題 A を 問題 B を定数時間で解けるアルゴリズムを用いて入力データサイズの多項式時間で解けることが保証できるとき，問題 B の NP 困難性が証明される．

4.2 最小トライアングルストリップ化問題が NP 困難であることの証明

NP 完全であることが既知である問題として，全ての頂点の次数が 3 の単純グラフ上でのハミルトン道問題 [4, 5] を考える（図 11）．

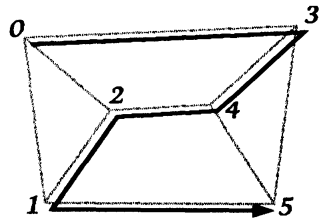


図 11: 全ての頂点の次数が 3 のグラフ上でのハミルトン道問題

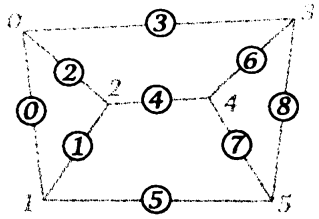
この問題に与えるグラフ G から以下のようにして，トライアングルリスト T を作成する．また，作成例を図 12 に示す．

Step 1. G 上の枝に番号を付け，T 上の頂点とする．

Step 2. G 上の頂点を，接続されている 3 つの枝に相当する頂点番号からなる T 上の三角形とする．

こうしてできた入力 T に対して，その最小トライアングルストリップ化の縮退回数が $2(n-1)$ 回（n は T の三角形数）になるかどうかという問題を考える．

この手順によって作られるトライアングルリスト T は，辺で接続する三角形の組を持たないため，1 縮退以下での三角形接続は不可能であるから，最低でも $2(n-1)$ 回の縮退が必要になり，$2(n-1)$ 回の縮退でトライアングルストリップ化できたとき（図 13），全ての三角形同士が 2 縮退で接続されていることになる．2 縮退で接続されている三角形同士は，共有する頂点 (G 上では枝) を持っているから，トライアングルストリップに現れる三角形出現順をそのまま，G 上の頂点列に書き換えれば，G 上のハミルトン道となる．



T :

t_0	0	2	3
t_1	0	1	5
t_2	1	2	4
t_3	3	6	8
t_4	4	6	7
t_5	5	7	8

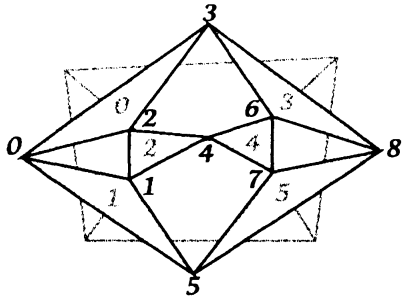


図 12: ハミルトン道問題への入力 G から，最小トライアングルストリップ化問題への入力 T を作る

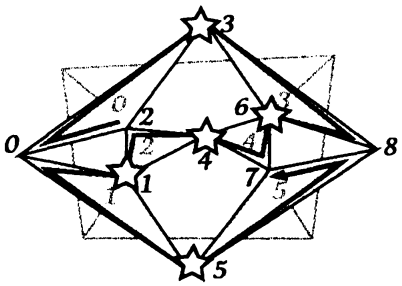


図 13: T 上の最小トライアングルストリップ化

一方，G 上にハミルトン道が存在する場合，連続する頂点同士は共有する枝（T 上では頂点）を持っていて，また 1 つ前の頂点と共有している枝と 1 つ後の頂点と共有している枝は別の枝のはずなので，ハミルトン道の順を三角形の出現順に置き換え，各三角形中の頂点出現順を 1 つ前の頂点と共有している枝が最初に，1 つ後の頂点と共有している枝が最後に来るように並びかえたとき，全ての三角形接続は 2 縮退になり，最小トライアングルストリップの長さは $2(n-1)$ となる．

よって，T 上の最小トライアングルストリップ化問題を解くことで，G 上のハミルトン道問題が解けることになり，T を作成するためには n に比例した時間しかかからないので，最小トライアングルストリップ化問題が NP 困難問題であることが証明された．

5 実用的な解法の構成

本節では，第 3 節で行った問題定義に対して変形を加えて，実用的に解きやすいような問題にすることを考える．変形の際に，問題の厳密性は失われてしまうが，元の問題の持つ性質を上手く残して良い解を求めることができる手法について考察していく．

5.1 問題の変形について

第 3 節で定義した問題は，第 4 節で言及したように NP 困難問題に属しているため，多項式時間では厳密に解くことができそうにないとされているが，それと同時に，実行可能解の数が非常に多く（三角形の並びかえが $n!$ 通り，各三角形中の頂点の並びかえが $(3!)^n$ 通りあり，$6^n n!$ 通りの実行可能解が存在）最適解を求める上での冗長性も高いため，局所探索等の発見的手法 [13] で解の改善をしていくことが難しい．

そこで，一旦発見的手法を効率良く適応できるような問題に変形してから，その問題に対して発見的手法を適応していくことにする．

5.2 マーク付けと評価の手法

問題の変形として，各三角形中の頂点出現順のみを先に評価するとう，マーク付けの問題に変形する手法を解説していく．各三角形中の頂点出現順のみを先に評価する場合，辺で接した三角形間の頂点出現順の関係を評価することが有効になる．また，関係評価に関して頂点出現順そのものを評価しようとすると，コストが複雑になり，前後入れ替えの冗長性も残るので，前後逆を同一視した頂点出現順を評価することにする．前後逆を同一視した頂点出現順は，3 頂点のうち真ん中に来る頂点を指定することにより決定できる．それを，各三角形中の 1 頂点にマークを付けることで表現する (図 14)．

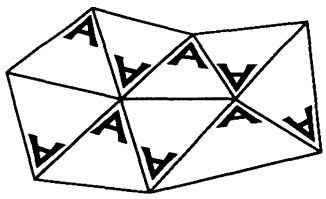


図 14: マーク付けの例．"A" の先側の頂点にマークが付いている

そして，辺の両側にある三角形間のマーク同士の関係にコストを与える．マークの関係は回転や反転したものを同一として，辺の片側にしか三角形が無いものも考慮すると，図 15 に表した 6 通りのどれかになる．それら 6 通りに次のようにコストを与える．

- Good!! で示した 3 つの関係はその関係の部分では縮退三角形を発生させないことができるので，コストは 0 を与える．
- Bad!! で示した 1 つの関係は，1 縮退によりつなぐことができるので，コストは 1 を与える．
- Terrible!! で示した 2 つの関係は，最低 1 つはストリップの切れ目を発生させてしまい 2〜4 縮退で，他の部分とつなぐことになる．ここでは 4 縮退の片側と見積もってコストは 2 を与える．(この関係のコストは 1 〜 2 間で適当に調整することも考えられる．)

まとめると，以下のような問題として考えられる．

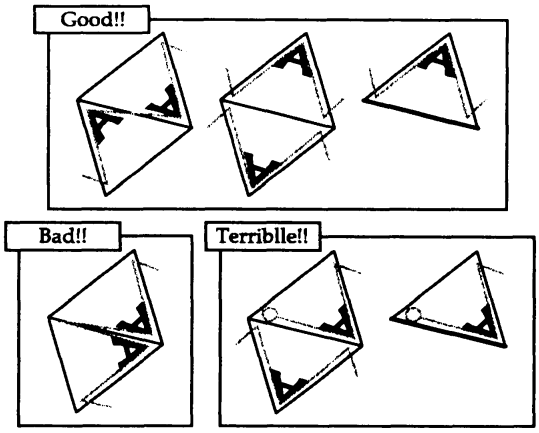


図 15: 辺に関するマークの関係とその評価

最小化 : 全ての辺の両側にある三角形のマーク同士の関係コストの合計
制約条件 : 各三角形中の 1 頂点にマークを付ける

5.3 マーク固定後のストリップ作成

三角形順を先に固定する方法と違い，マーク付けを固定した条件下での最小トライアングルストリップの導出を厳密に行うのは簡単ではないと予想される．0, 1 縮退でつなぐことのできるマーク関係でも，そのつながり関係がループしていた場合どこかは切断しなければならないし，2, 3 縮退でつなげられる候補が複数存在し，どこにつなげば最も得かが分かりにくいこともある．

しかし，マークのコストが小さい場合，全体でみれば大きい差にはならないので，例えば 0, 1, 2, 3, 4 の優先順で接続できる箇所を発見順につないでしまう等の手法でもある程度良い解は導出する事ができる．

5.4 元の問題とのギャップ

マーク付けの手法に関しては，評価時点では 2, 3, 4 縮退の区別をしない．このため 2, 3 縮退でうまくつないでいけるケースを悪く評価してしまう．また，0, 1 縮退のループが発生しているケースは，どこかを切断しなければならないので，実際より良く評価してしまう．トライアングルストリップの端の扱いに関しても多少ギャップが発生するが，やはり最大でも 4 縮退の差にしかならないのでほぼ無視できる．

6 簡単な実装実験

6.1 実験の目的

第 5 節で提案したマーク付けのヒューリスティクスの有効性を示すため，簡単な実験を行った．

6.2 実装したアルゴリズム

マーク付けに対するアルゴリズムは，以下のようなものを使った．

Step 1. マーク済み三角形の集合 M を定義，乱数で選んだ初期三角形 1 つに乱数でマークを付け M に追加する．

Step 2. M に含まれていない M に接続されている三角形を乱数で選び（この際，マークが付いてる頂点と逆側の辺で隣接している三角形は選ばれる確率を落とす）図 16 のように隣接部分のマークのコストが 0 になるようにマークを付け，M に追加する．

Step 3. Step 2. を全ての三角形にマークが付くまで繰り返す．

Step 4. 三角形毎にマークを付け替えて，全体のコストが改善するかどうか調べ，改善する場合マークを付け替える．(図 17)

Step 5. Step 4. を数周繰り返す．

Step 6. Step 1〜5 を数回繰り返し，最も結果が良いものを選択する．

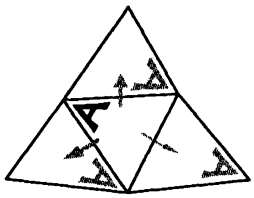


図 16: 隣接三角形へのコストの低いマーク付け

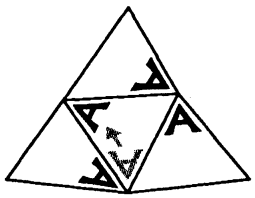


図 17: 局所的なマークの改善

マークからのストリップ生成は，縮退が少ない接続から順につないでいくだけの手法をとった．ループが発生する状況の場合も，発見順につないでしまう．4 縮退で跳ぶ場合に関しては，どこをつないでもストリップ長自体は同じになるが，図の見やすさを考慮し，適当に頂点座標の近い所をつないでいる．

6.3 実験結果

Stanford Bunny に対して提案するアルゴリズムを適用し NvTriStrip[9], Tri Stripper[3] の 2 つの既存手法と比較した．実験結果を表 2, 図 18〜図 19 に示す．

基本的にモデルの位相データに必要な頂点列の大きさで比較を行った．NvTriStrip は提案手法と同様に 1 本のトライアングルストリップを出力するので，そのままストリップ長を記した．Tri Stripper に関しては，1 本のつながったトライアングルストリップを出力するのでなく，複数のストリップとトライアングルリストを出力する形式なので，ストリップはこちらでつないで，リストはそのまま足して計算している．提案手法に関しては繰り返し回数を増やすことによって，ある程度改善する可能性があるため，20 回繰り返し時の結果と 2000 回繰り返し時の結果を載せている．実行時間に関しては，提案手法の 2000 回繰り返しも含めてどの手法も 10 分以内程度で終了したので，特に比較はしていない．既存手法は両者とも頂点キャッシュの意識が強いため，フェアな比較とはいえないが，ストリップ長に関しては良い結果を出せることが予想できる．

図 18〜図 19 はマーク付けによって作られたトライアングルストリップを，ストリップの接続がされていないところに隙間を作る形で可視化した．ストリップ上の頂点座標をストリップの前後の頂点の座標で引っ張った形なので，本来縮退が起こっている部分も三角形として描画されている．図 19 のように平面上に綺麗に三角形が並べられている場合に，それに沿った綺麗なストリップがしっかり出ていることが見てとれる．

表 2: Stanford Bunny (頂点数 : 34834, 三角形数 : 69451) に対するトライアングルストリップ化の結果

	頂点列長	triangle list
トライアングルリスト	208353	100%
NvTriStrip	103015	49.4%
Tri Stripper	92893	44.6%
提案手法 (マーク付け)	85071 ～ 85409	40.8 ～ 41.0%
縮退 0 とした値 (下界値)	69453	33.3%

7 まとめ

ポリゴンモデル情報の良いトライアングルストリップ化を行うために，最小トライアングルストリップ問題について組合せ最適化問題として考察し，マーク付け問題に変形することによって，トライアングルストリップのストリップ長に関しては良い結果を出せることが分かった．各論での課題を挙げると，計算複雑性に関してはより狭い範囲の問題 (全ての三角形が辺接続等) の NP 困難性の証明が課題になっており，実用的解法の構成については，理論的裏付けを進めていく必要がある．実装面では，比較検証をより多くのサンプルで行っていくことや，より良いストリップが得られるよう，アルゴリズムを改善すること等が挙げられる．

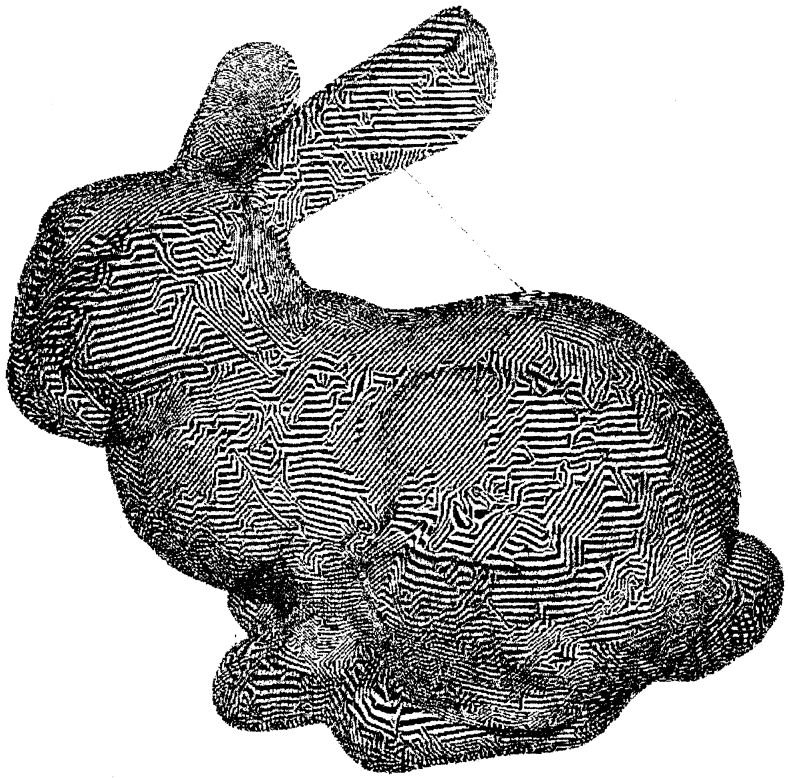


図 18: マーク付けによるトライアングルストリップの可視化 (Stanford Bunny) : 全体

参 考 文 献

[1] Deering,M., "Geometry Compression", *Computer Graphics, Proceedings of the ACM SIGGRAPH '95*, pp.13-20(1995).

[2] Evans,F., S.Skiena, and A.Varshney, "Optimizing Triangle Strips for Fast Rendering", *Proceedings IEEE Visualization '96*, pp.319-326(1996).

[3] Fautre,T., "GPSnoopy's Development Arena", (http://users.pandora.be/tfautre/softdev/tristripper/).

[4] Garay,M.R., D.S.Johnson, and R.Endre Tarjan, "The Planar Hamiltonian Circuit Problem is NP-Complete", *SIAM Journal on Computing*, vol.5, no.4, pp.704-714(1976).

[5] Garay,M.R., and D.S.Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, WH Freeman & Co.(1979).

[6] Gopi,M., and David Eppstein, "Single-Strip Triangulation of Manifolds with Arbitrary Topology", *Computer Graphics Forum*, vol.23, no.3, pp.371-379(2004).

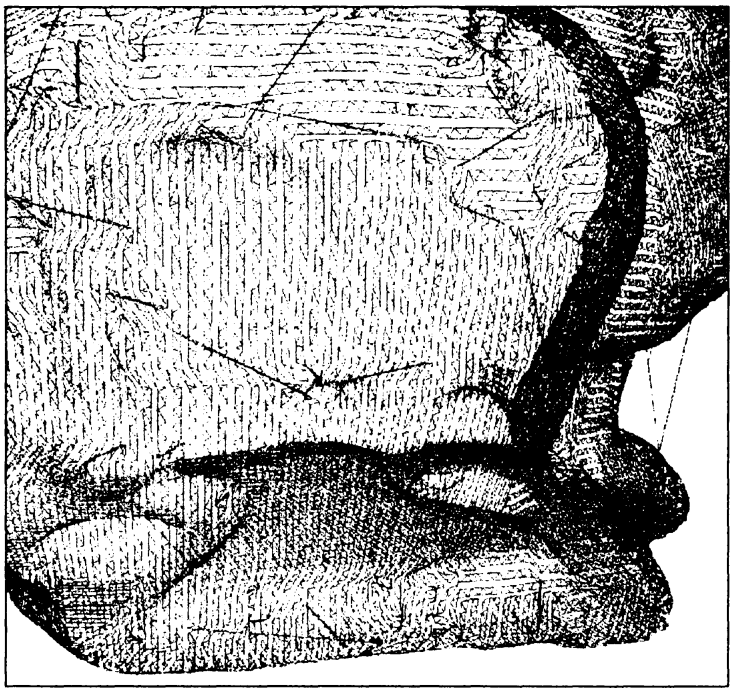


図 19: マーク付けによるトライアングルストリップの可視化 (Stanford Bunny) : 平坦に近い部分

[7] Hoppe,H., "Optimization of Mesh Locality for Transparent Vertex Caching", *Computer Graphics, Proceeding of the ACM SIGGRAPH '99*, pp.269-276(1999).

[8] 茨木俊秀, 組合せ最適化 : 分枝限定法を中心として, 産業図書 (1983).

[9] NVIDIA, "NvTriStrip Library", (http://developer.nvidia.com/object/nvtristrip_library.html).

[10] 大川善邦・大澤文孝・登 大遊・成田拓郎, DirectX9 実践プログラミング, 工学社 (2003).

[11] Stanford Computer Graphic Laboratory, "Stanford Computer Graphics Laboratory", (http://graphics.stanford.edu/)

[12] 床井浩平, GLUT による OpenGL 入門, 工学社 (2005).

[13] 柳浦陸憲・茨木俊秀, 組合せ最適化 : メタ戦略を中心として, 朝倉書店 (2001).